CLAIMS

1-29. (cancelled)

30. (currently amended) A method for analyzing a scalability of an application server, comprising:

providing a test configuration, wherein the test configuration includes a component quantity generated using a component template, a jar quantity for containing the component quantity, a node quantity, a server quantity, a resource quantity, and resource types;

creating an application based on the test configuration, wherein the application includes a quantity of components that match the provided component quantity and a quantity of jars that match[[es]] the provided jar quantity;

deploying, installing, and starting the application on the application server, wherein the application server is implemented on a node, comprising a computerized device, that is part of a hierarchy of nodes that match the provided node quantity, and wherein the application is deployed using a deploy tool of the application server;

creating a client program based on the test configuration, wherein the client program simulates use of the application by a quantity of clients that match the provided component quantity;

running the client program against the application by conducting a database transaction;

monitoring performance metrics during the deploying, installing, starting and running steps to verify the scalability of the application server, the monitoring comprising monitoring of:

deployment metrics selected from the group consisting of an enterprise archive file size, a heap size, a memory consumption, a processor usage, and an application deploy time during the deploying step;

monitoring installation metrics selected from the group consisting of a memory consumption, a processor usage, and an application install time during the installing step;

monitoring start metrics selected from the group consisting of a memory increase, a processor usage, and an application start time during the starting step; and

2

monitoring client metrics selected from the group consisting of a memory consumption, a processor usage, a transaction time, and an application run time during the running step;

and

generating output based on the performance metrics, wherein the output includes a comparison of the performance metrics for the provided component quantity and the provided jar quantity, versus a potential performance metric for the application with at least one of a different quantity of components and a different quantity of jars.

31. (currently amended) A computer-implemented system for analyzing a scalability of an application server, comprising:

at least one processing unit;

a memory operably associated with the at least one processing unit; and

a scalability analysis system storable in the memory[[.]] and executable by the at least one processing unit, the scalability analysis system comprising:

a test configuration system for providing a test configuration, wherein the test configuration includes a component quantity generated using a component template, a jar quantity for containing the component quantity, a node quantity, a server quantity, a resource quantity, and resource types, and wherein the test configuration system provides a graphical user interface for inputting the test configuration;

an application generation system for creating an application based on the test configuration, wherein the application includes a quantity of components that match the provided component quantity and a quantity of jars that match[[es]] the provided jar quantity, wherein the application is deployed, installed and started on the application server, and wherein the application server is implemented on a node, comprising a computerized device, that is part of a hierarchy of nodes that match the provided node quantity, and wherein the application is deployed using a deploy tool of the application server;

a client program system for creating a client program based on the test configuration, and for running the client program against the application by conducting a database transaction, wherein the client program simulates use of the application by a quantity of clients that match the provided component count;

3

a metric monitoring system for monitoring performance metrics while the application is deployed, installed, and started and when the client program is run to verify the scalability of the application server, wherein the metric monitoring system monitors:

deployment metrics selected from the group consisting of an enterprise archive file size, a heap size, a memory consumption, a processor usage, and an application deploy time as the application is being deployed;

installation metrics selected from the group consisting of a memory consumption, a processor usage, and an application install time as the application is being installed;

start metrics selected from the group consisting of a memory increase, a processor usage, and an application start time as the application is being started; and

client metrics selected from the group consisting of a memory consumption, a processor usage, a transaction time, and an application run time as the client program runs;

and

an output system for generating output based on the performance metrics, wherein the output includes a comparison of illustrates the performance metrics for the provided component quantity and the provided jar quantity, versus a potential performance metric for the application with at least one of a different quantity of components and a different quantity of jars.

32. (currently amended) A non-transitory computer-readable storage- medium storing computer instructions which when executed, enable[[s]] a computer system to analyze a scalability of an application server, the computer instructions comprising :

providing a test configuration, wherein the test configuration includes a component quantity generated using a component template, a jar quantity for containing the component quantity, a node quantity, a server quantity, a resource quantity, and resource types;

creating an application based on the test configuration, wherein the application includes a quantity of components that match the provided component quantity and a quantity of jars that match the provided jar quantity;

deploying, installing, and starting the application on the application server, wherein the application server is implemented on a node, comprising a computerized device, that is part of a hierarchy of

4

nodes that match the provided node quantity, and wherein the application is deployed using a deploy tool of the application server;

creating a client program based on the test configuration, wherein the client program simulates use of the application by a quantity of clients that match the provided component quantity;

running the client program against the application by conducting a database transaction;

monitoring performance metrics during the deploying, installing, starting and running steps to verify the scalability of the application server, the monitoring comprising monitoring of:

deployment metrics selected from the group consisting of an enterprise archive file size, a heap size, a memory consumption, a processor usage, and an application deploy time during the deploying step;

monitoring installation metrics selected from the group consisting of a memory consumption, a processor usage, and an application install time during the installing step;

monitoring start metrics selected from the group consisting of a memory increase, a processor usage, and an application start time during the starting step; and

monitoring client metrics selected from the group consisting of a memory consumption, a processor usage, a transaction time, and an application run time during the running step;

and

generating output based on the performance metrics, wherein the output includes a comparison of the performance metrics for the provided component quantity and the provided jar quantity, versus a potential performance metric for the application with at least one of a different quantity of components and a different quantity of jars.

5